Retrieve clinical trial information

Ralf Herold

2024-01-14

Get started

Attach package `ctrdata`

```
library(ctrdata)
citation("ctrdata")
```

Remember to respect the registers' terms and conditions (see `ctrOpenSearchPagesInBrowser(copyright = TRUE)`). Please cite this package in any publication as follows: Ralf Herold (2024). ctrdata: Retrieve and Analyze Clinical Trials in Public Registers. R package version 1.16.0. https://cran.r-project.org/package=ctrdata

Open register's advanced search page in browser

These functions open the browser, where the user can start searching for trials of interest.

```
# Please review and respect register copyrights:
ctrOpenSearchPagesInBrowser(
  copyright = TRUE
)
# Open browser with example search:
ctrOpenSearchPagesInBrowser(
  url = "cancer&age=under-18",
  register = "EUCTR"
)
```

Adjust search parameters and execute search in browser

Refine the search until the trials of interest are listed in the browser. The total number of trials that can be retrieved with package `ctrdata` is intentionally limited to queries with at most 10000 result records.

Copy address from browser address bar to clipboard

Use functions or keyboard shortcuts according to the operating system. See here for our automation to copy the URLs of a user's queries in any of the supported clinical trial registers.

Get address from clipboard

The next steps are executed in the R environment:

```
q <- ctrGetQueryUrl()
# * Using clipboard content as register query URL: https://www.clinicaltrialsregister.eu/ctr-search/sea
# * Found search query from EUCTR: query=cancer&age=under-18&status=completed&phase=phase-one

q
#                                                  query-term  query-register
# 1 query=cancer&age=under-18&status=completed&phase=phase-one           EUCTR

# To check, this opens a browser with the query
ctrOpenSearchPagesInBrowser(url = q)
```

Retrieve protocol-related information, transform, save to database, check

```
# Count number of trial records
ctrLoadQueryIntoDb(
  queryterm = q,
  only.count = TRUE
)$n
# * Found search query from EUCTR: query=cancer&age=under-18&status=completed&phase=phase-one
# * Checking trials in EUCTR...
# Retrieved overview, multiple records of 97 trial(s) from 5 page(s) to be downloaded (estimate: 5 MB)
# [1] 97

# Connect to a database and chose a table / collection
db <- nodbi::src_sqlite(
  dbname = "sqlite_file.sql",
  collection = "test"
)

# Retrieve records, download into database
ctrLoadQueryIntoDb(
  queryterm = q,
  con = db
)
# * Found search query from EUCTR: query=cancer&age=under-18&status=completed&phase=phase-one
# * Checking trials in EUCTR...
# Retrieved overview, multiple records of 97 trial(s) from 5 page(s) to be downloaded (estimate: 5 MB)
# (1/3) Downloading trials...
# Note: register server cannot compress data, transfer takes longer (estimate: 20 s)
# Download status: 5 done; 0 in progress. Total size: 7.91 Mb (100%)... done!
# (2/3) Converting to NDJSON (estimate: 2 s)...
# (3/3) Importing records into database...
# = Imported or updated 377 records on 97 trial(s)
# No history found in expected format.
# Updated history ("meta-info" in "test")
# $n
# [1] 377
```

```
# Show which queries have been downloaded into database
dbQueryHistory(con = db)
#       query-timestamp query-register query-records
# 1 2024-01-14 12:40:16          EUCTR           377
#                                                 query-term
# 1 query=cancer&age=under-18&status=completed&phase=phase-one
```

With a file-base SQLite database, this takes about 20 seconds for about 300 records, with most of the time needed for internet-retrieval with is slow from this register. Speed is higher with other registers, with using MongoDB and with memory-based SQLite.

Repeat and update a previous query

Instead of "last", an integer number can be specified for `querytoupdate` that corresponds to the number when using `dbQueryHistory()`.

```
ctrLoadQueryIntoDb(
  querytoupdate = "last",
  con = db
)
```

Depending on the register, an update (differential update) is possible or the original query is executed fully again.

Retrieve results

For EUCTR, result-related trial information has to be requested to be retrieved, because it will take longer to download and store. For CTGOV, any results are always included in the retrieval. Note that trial documents, including any results reports, can be downloaded by specifying parameter `documents.path`, see `help(ctrLoadQueryIntoDb)`.

```
ctrLoadQueryIntoDb(
  queryterm = q,
  euctrresults = TRUE,
  con = db
)
# * Found search query from EUCTR: query=cancer&age=under-18&status=completed&phase=phase-one
# * Checking trials in EUCTR...
# Retrieved overview, multiple records of 97 trial(s) from 5 page(s) to be downloaded (estimate: 5 MB)
# (1/3) Downloading trials...
# Note: register server cannot compress data, transfer takes longer (estimate: 20 s)
# Download status: 5 done; 0 in progress. Total size: 7.91 Mb (100%)... done!
# (2/3) Converting to NDJSON (estimate: 2 s)...
# (3/3) Importing records into database...
# = Imported or updated 377 records on 97 trial(s)
# * Checking results if available from EUCTR for 97 trials:
# (1/4) Downloading and extracting results (. = data, F = file[s] and data, x = none):
# Download status: 97 done; 0 in progress. Total size: 48.64 Mb (100%)... done!
# Download status: 21 done; 0 in progress. Total size: 84.53 Kb (308%)... done!
# Download status: 21 done; 0 in progress. Total size: 84.53 Kb (308%)... done!
# Download status: 21 done; 0 in progress. Total size: 84.53 Kb (308%)... done!
```

```
# F . . . . F . . F . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
# (2/4) Converting to NDJSON (estimate: 8 s)...
# (3/4) Importing results into database (may take some time)...
# (4/4) Results history: not retrieved (euctrresultshistory = FALSE)
# = Imported or updated results for 76 trials
# Updated history ("meta-info" in "test")
```

With a file-base SQLite database, this takes about 4 minutes for about 300 records, with most of the time needed for merging result- and protocol-related information in SQLite; this is much faster with MongoDB and PostgreSQL.

The download or presence of results is not recorded in `dbQueryHistory()` because the availability of results increases over time.

Add information from another register

The same collection can be used to store (and analyse) trial information from different registers. Example:

```
ctrLoadQueryIntoDb(
  queryterm = "https://classic.clinicaltrials.gov/ct2/results?cond=neuroblastoma&recrs=e&age=0&intr=Drug
  con = db
)
# * Appears specific for CTGOV CLASSIC
# * Found search query from CTGOV: cond=neuroblastoma&recrs=e&age=0&intr=Drug
# * Checking trials in CTGOV classic...
# Retrieved overview, records of 216 trial(s) are to be downloaded (estimate: 1.7 MB)
# (1/3) Downloading trial file...
# Download status: 1 done; 0 in progress. Total size: 1.52 Mb (100%)... done!
# (2/3) Converting to NDJSON (estimate: 10 s)...
# (3/3) Importing records into database...
# = Imported or updated 216 trial(s)
# Updated history ("meta-info" in "test")
```

With a file-base SQLite database, this takes about 10 seconds for about 200 records.

Note that in this example, a warning message may be issued from importing an NDJSON file with trial records. The warning arises from the high level of complexity of some of the XML content of some of the trial records. The issue can be resolved by increasing in the operating system the stack size available to R, see: https://github.com/rfhb/ctrdata/issues/22

Add records from CTIS register into the same collection

As of April 2023, more than 160 trials are publicly accessible in CTIS, which has to be used since January 2023 for new clinical trials in the EU. Queries in the CTIS search interface can be automatically copied to the clipboard so that a user can paste them into `queryterm`, see here.

```
# Retrieve trials from another register:
ctrLoadQueryIntoDb(
  queryterm = "ageGroupCode=3",
  register = "CTIS",
  con = db
)
```

Add personal annotations

When downloading trial information, the user can specify an annotation to all records that are downloaded. By default, annotations are accumulated if trial records are loaded again or updated; alternatively, annotations can be replaced.

Annotations are useful for analyses, for example to specially identify subsets of records in the database.

```
ctrLoadQueryIntoDb(
  queryterm = "https://classic.clinicaltrials.gov/ct2/results?cond=neuroblastoma&recrs=e&age=0&intr=Drug
  annotation.text = "site_DE ",
  annotation.mode = "append",
  con = db
)
# * Found search query from CTIS: ageGroupCode=3
# * Checking trials in CTIS...
# (1/5) Downloading trials list . . found 392 trials
# (2/5) Downloading and processing part I and parts II... (estimate: 80 Mb)
# Download status: 392 done; 0 in progress. Total size: 62.56 Mb (100%)... done!
# Download status: 2 done; 0 in progress. Total size: 30 b (100%)... done!
# Download status: 2 done; 0 in progress. Total size: 30 b (100%)... done!
# Download status: 2 done; 0 in progress. Total size: 30 b (100%)... done!
# . . . . . .
# (3/5) Downloading and processing additional data:
# publicevents, summary, layperson, csr, cm, inspections, publicevaluation (estimate: 20 Mb)
# Download status: 680 done; 0 in progress. Total size: 17.83 Mb (100%)... done!
# 390
# (4/5) Importing records into database...
# (5/5) Updating with additional data: . . . . .
# = Imported / updated 392 / 390 / 390 / 4 / 1 / 3 records on 392 trial(s)
# Updated history ("meta-info" in "test")
# $n
# [1] 392
```

Add information using trial identifiers

When identifiers of clinical trials of interest are already known, this example shows how they can be processed to import the trial information into a database collection. This involves constructing a query that combines the identifiers and then iterating over the sets of identifiers. Note to combine identifiers using "+OR+" into the `queryterm`, and that `register` has to be specified.

```
# ids of trials of interest
ctIds <- c(
  "NCT00001209", "NCT00001436", "NCT00187109", "NCT01516567", "NCT01471782", "NCT03042429",
  "NCT00357084", "NCT00357500", "NCT00365755", "NCT00407433", "NCT00410657", "NCT00436657",
  "NCT00436852", "NCT00445965", "NCT00450307", "NCT00450827", "NCT00471679", "NCT00486564",
  "NCT00492167", "NCT00499616", "NCT00503724", "NCT00509353", "NCT00520936", "NCT00536601",
  "NCT00567567", "NCT00578864", "NCT00601003", "NCT00644696", "NCT00646230", "NCT00659984",
  "NCT00716976", "NCT00743496", "NCT00793845", "NCT00806182", "NCT00831844", "NCT00867568",
  "NCT00877110", "NCT00885326", "NCT00918320", "NCT00923351", "NCT00939770", "NCT00960739"
)

# split into sets of each 25 trial ids
```

```
# (larger sets e.g. 50 may still work)
idSets <- split(ctIds, ceiling(seq_along(ctIds) / 25))

# variable to collect import results
result <- NULL

# iterate over sets of trial ids
for (idSet in idSets) {

  # import
  setResult <- ctrLoadQueryIntoDb(

    # for CTGOV classic, EUCTR, ISRCTN use:
    # queryterm = paste0(idSet, collapse = "+OR+"),

    # for CTGOV use:
    queryterm = paste0("https://www.clinicaltrials.gov/search?term=",
                       paste0(idSet, collapse = " ")),

    # specify register that holds the information
    # register = "CTGOV",
    con = db
  )

  # append results
  result <- c(result, list(setResult))
}

# inspect results
as.data.frame(do.call(rbind, result))[, c("n", "failed")]
#    n failed
# 1 25   NULL
# 2 17   NULL
```

Note that from `CTIS`, trial information *identified by trial identifiers* can be retrieved only one-by-one, repeating queries for each trial such as https://euclinicaltrials.eu/app/#/search?number=2023-503994-39-00.

Find synonyms of active substance names

Not all registers automatically expand search terms to include alternative terms, such as codes and other names of active substances. To obtain a character vector of synonyms for any active substance name, use:

```
ctrFindActiveSubstanceSynonyms(
  activesubstance = "imatinib"
)
# [1] "imatinib"  "gleevec" "sti 571" "glivec" "CGP 57148" "st1571"
```

These names can then be used in queries in any register.

Using a MongoDB database

This example works with a free service here. Note that the user name and password need to be encoded. The format of the connection string is documented at https://docs.mongodb.com/manual/reference/connection-string/.

For recommended databases, see vignette Install R package ctrdata.

```
# Specify base uri for remote MongoDB server,
#  as part of the encoded connection string
db <- nodbi::src_mongo(
  # Note: this provides read-only access
  url = "mongodb+srv://DWbJ7Wh:bdTHh5cS@cluster0-b9wpw.mongodb.net",
  db = "dbperm",
  collection = "dbperm")

# Since the above access is read-only,
# just obtain fields of interest:
dbGetFieldsIntoDf(
  fields = c("a2_eudract_number",
             "e71_human_pharmacology_phase_i"),
  con = db)
#                   _id a2_eudract_number e71_human_pharmacology_phase_i
# 1 2010-024264-18-3RD    2010-024264-18                            TRUE
# 2  2010-024264-18-AT    2010-024264-18                            TRUE
# 3  2010-024264-18-DE    2010-024264-18                            TRUE
# 4  2010-024264-18-GB    2010-024264-18                            TRUE
# 5  2010-024264-18-IT    2010-024264-18                            TRUE
# 6  2010-024264-18-NL    2010-024264-18                            TRUE
```